reconstructed representations (e.g., point clouds, truncated sign distance function (TSDF), populated voxels, mesh, light fields, etc.) can be associated with data processing levels or requirements and/or display characteristics. For example, a TSDF representation can be assigned a processing level greater than a level for point clouds. At block **1206**, current data processing and/or display capabilities of the receiving system can be matched to the processing levels or requirements and/or display characteristics of the representations to select a representation type to use. In some implementations where process **1201** reconstruction is performed on a system other than the receiving system (e.g., in versions **602**-**610** of FIG. **6**) representations can also be characterized according to an average size of that type of reconstruction and the type of reconstruction selected can further be selected based on available bandwidth.

[0110] While, as discussed above, in various implementations, any block from any of the flow charts can be removed or rearranged, blocks **1204** and **1206** are illustrated in broken lines to call out specific implementations where blocks **1204** and **1206** may or may not be performed. In some implementations, there is no dynamic reconstruction type selection and thus process **1201** can go from block **1202** to block **1208**.

[0111] At block **1208**, process **1201** can create a 3D representation from the decompressed data, in either a default representation format or a format selected at block **1206**. Reconstruction block **1201** can accomplish this transformation of the depth data into the 3D representation using the calibration data to combine data from multiple sources and/or transform the captured data into position and contour information in 3D space. For example, each pixel in a depth image depicting a user can be transformed into a 3D representation of at least part of the user by applying transformations based on the intrinsic and extrinsic calibration parameters of the camera. The transformations can take each pixel taken at the camera location and determine a corresponding point in 3D space representing a point on the surface of the user. In some implementations, the reconstruction stage can apply shading or color data to the 3D representation, using the calibration data to map portions of the shading or color data to the 3D representation. In various implementations, creating a 3D representation can be performed using, e.g., an Integrated 6DoF Video Camera and System Design (see e.g., the system described at https://research.fb.com/publications/an-integrated-6dof-video-camera-and-system-design/ and https://research.fb.com/wp-content/uploads/2019/09/An-Integrated-6DoF-Video-Camera-and-System-Design.pdf); and Immersive Light Field Video with a Layered Mesh Representation sysetm (see e.g., the system described at https://augmentedperception.github.io/deepviewvideo/ and https://storage.googleapis.com/immersive-lf-video-siggraph2020/Immersive-LightFieldVideoWithALayeredMeshRepresentation.pdf); a Dynamic Fusion system (see e.g., the system described at https://grail.cs.washington.edu/projects/dynamicfusion/ and https://grail.cs.washington.edu/projects/dynamicfusion/papers/DynamicFusion.pdf); or a Fusion4D: system (see e.g., the system described at https://www.microsoft.com/en-us/research/wp-content/uploads/2016/11/a114-dou.pdf); each of these is incorporated herein by reference.

[0112] In some cases, part or all of the 3D representation may not be direct translations of captured data, such as where none of the captured images of the sending user depict a portion of the 3D representation or where bandwidth or processing limitations did not allow sufficient time to provide a portion of the images depicting the sending user. In some such cases, some or all of the 3D representation can be approximated with an avatar representation of the sending user, a machine learning estimation of the missing portion(s), or previously captured versions of the missing portion(s) of the sending user. In other such cases, the reconstruction stage can be skipped, reverting instead to traditional 2D video calling or voice calling.

[0113] FIG. **13** is a flow diagram illustrating a process **1301** (e.g., the process performed by block **514** of FIG. **5**) used in some implementations of the present technology for a render stage of a 3D conversation pipeline. At block **1302**, process **1301** can receive a 3D representation from process **1201** and an indication of a current viewpoint of the receiving user. At block **1304**, process **1301** can determine whether process **1301** is being performed to dynamically select rendering settings specific to the receiving system. For example, where the render stage is performed on a system other than the receiving system (e.g., versions **602**, **604**, and **608** of FIG. **6**), the render stage can receive display capabilities (e.g., resolution, frame rate, single display or stereo display, etc.) of each receiving system at block **1306** and can use those specifics to, at block **1308**, render one or more images to those capabilities. For example, images can be rendered to the same resolution as a receiving system or an image can be rendered for each eye viewpoint of the receiving user. While, as discussed above, in various implementations, any block from any of the flow charts can be removed or rearranged, blocks **1304** and **1306** are illustrated in broken lines to call out specific implementations where blocks **1304** and **1306** may or may not be performed. In some implementations, there is no dynamic, receiver-specific rendering and thus process **1301** can go from block **1302** to block **1308**.

[0114] At block **1308**, process **1301** can render one or more images for display by a receiving system using default render parameters (either generic parameters or previously configured for the particular receiving system) or parameters selected for the receiver specifics obtained at block **1306**. Generating these images can include placing a virtual camera, in relation to the 3D representation, at the receiving user's viewpoint (or two virtual cameras at the viewpoint of each eye of the receiving user) to produce one or two 2D images from the 3D representation from the viewpoint of the virtual camera(s). Where the reconstruction stage did not include adding color data to the 3D representation, the render stage can further include adding color data, captured by traditional cameras and transformed according to the calibration data, to the rendered images. In some implementations, this can include using color data from the camera closest to the viewpoint of the receiving user. In some implementations, rendering can include performing additional modifications, such as removing an artificial reality device from the display of the sending user (see e.g., FIG. **15**), adding additional feeds (e.g., a presentation or slide deck to the images), adding stickers, effects or filters, etc.

[0115] FIG. **14** is a flow diagram illustrating a process **1401** (e.g., the process performed by block **516** of FIG. **5**) used in some implementations of the present technology for a display stage of a 3D conversation pipeline. At block **1402**, process **1401** can receive the 2D images rendered by process **1401** (or 3D representations if the receiving system has a